A NetFlow/IPFIX implementation with OpenFlow

José Suárez-Varela[1] **, Pere Barlet-Ros[1], and Valentín Carela-Español[2]

[1] UPC BarcelonaTech, Spain,
{jsuarezv,pbarlet}@ac.upc.edu
[2] Talaia Networks, S.L., Spain,
vcarela@talaia.io

The paradigm of Software-Defined networking (SDN) has recently gained lots of attention from research and industry. The logically centralized control plane provides flexibility and enables to perform a fine-grained management of the network, taking advantage of the decision making from a global perspective of the network. To be successful in dynamic environments, monitoring takes a key role in SDN given that management applications often need to make use of accurate and timely traffic measurements at different aggregation levels.

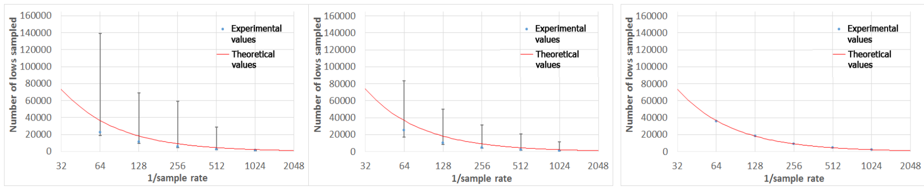
In this work we propose a solution for SDN to emulate NetFlow in traditional networks. In this way, traffic statistics are aggregated separately into flow records and are reported to a colector when a timeout expires. To achieve it, we use only OpenFlow [1] messages and capabilities, as it has become the de facto protocol for the communication between control and data planes in SDN.

An inherent issue of SDN is its scalability. To address it, a common practice in traditional networks is to implement traffic sampling when collecting flow measurements. As for the sampling schemes, two different approaches can be mainly distinguished: packet sampling and flow sampling. In this work, we implement flow sampling because it can be provided without requiring modifications to the OpenFlow specification. In addition, several studies have shown that packet sampling it is not the most adequate solution for some fine-grained monitoring applications [2]. This is particularly the case of applications like traffic classification or anomaly detection, where flow sampling can be a better alternative.

We propose three different methods to perform flow sampling depending on the OpenFlow capabilities available in the switch. We assume that the switches have support for OpenFlow 1.1.0 and later versions so, they have at least support for multiple tables. However, our solution can be adapted for the use of OpenFlow 1.0.0 with some limitations.

The first method performs flow sampling based on matches of pairs of IP addresses suffixes. In this way, we set the length of the IP suffixes to control the sampling rate. The second method is based on matches of pairs of ports. In this case, to perform flow sampling as random as possible, we choose randomly n ports out of 65,535, which is the total number of possible ports (port fields

** This work was done in collaboration with Talaia Networks under the framework of the H2020 SME Instrument Phase 2 project "SDN-Polygraph" (ref. 726763). This work was also supported by the Spanish Ministry of Economy and Competitiveness and EU FEDER under grant TEC2014-59583-C2-2-R (SUNSET project) and by the Catalan Government (ref. 2014SGR-1427).



(a) IP-based sampling (b) Port-based sampling (c) Hash-based sampling

Fig. 1: Evaluation of sampling rate

have 16 bits). The last method consists of computing a hash function on the traditional 5-tuple fields of the packet header and selecting it if the hash value falls in a particular range. This method much better controls the sampling rate, since we can assume that a hash function is homogeneous along its whole range for all the different flows in the switch. However, this method requires to support group tables (optionally available from OpenFlow 1.1.0) with *select* buckets and having an accurate algorithm in the switch, which is external to the OpenFlow specification, to balance the load properly among buckets.

The three proposed methods are evaluated in our monitoring solution using the well-known SDN controller OpenDaylight [3]. Firstly, we analyze the accuracy of each of these methods, and then, we evaluate the overhead contribution of our monitoring system. We conduct experiments in a small testbed with an Open vSwitch [4], a host (VM host) which injects traffic into the switch and another host which acts as a sink for all the traffic forwarded. All the experiments make use of a real traffic trace from CAIDA [5] which was filtered to keep only the TCP and UDP traffic. This traffic contains 2,353,413 different flows and was captured from a 10 Gbps link of a data center in Chicago in Frebruary 2016.

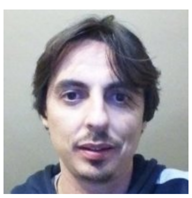
To analyze the accuracy applying the sampling rate, we evaluate the number of flows sampled for each of the three methods and compare it with the theoretical number of flows if we used a completely random selection function. We show in Fig. 1 the results for the three methods. We can see that the median values obtained are quite close to the theoretical values, especially in the hash-based method. It means that in the average case, these methods apply properly the sampling rate established. However, we can see that the IP and the port-based techniques present a high variability between experiments. In other words, depending on the IP suffixes or sets of ports selected, we can over- or under-sample. Given these results, we can asseverate that the method which better works is the hash-based one. As for the sampling methods based on IP suffixes and ports, we infer that they can achieve good results, but it is recommended a previous analysis of the traffic in the network to choose properly the IP suffixes or sets of ports to be sampled.

In this work we presented a monitoring solution fully compliant with OpenFlow which emulates the NetFlow/IPFIX operation. We propose three sampling methods that can be implemented in current switches without requiring any modification to the OpenFlow specification. We implemented them in OpenDaylight and evaluated their accuracy in a testbed with real traffic. As future work, we plan to implement smarter algorithms to retrieve the statistics more accurately and also a packet sampling method, although we find it more challenging.

References

1. Mckeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: OpenFlow: enabling innovation in campus networks. In ACM SIGCOMM CCR, 38, 69-74 (2008)
2. Sekar, V., Reiter, M. K., Zhang, H.: Revisiting the case for a minimalist approach for network flow monitoring. In ACM IMC (2010)
3. OpenDaylight project. `http://www.opendaylight.org/`
4. Pfaff, B., Pettit, J., Amidon, K., Casado, M., Koponen, T., Shenker, S.: Extending Networking into the Virtualization Layer. In Hotnets (2009)
5. The CAIDA Anonymized Internet Traces 2016 Dataset. `http://www.caida.org/data/passive/passive_2016_dataset.xml`

José Suárez-Varela received a B.Sc. degree in telecommunications engineering from the University of Granada (UGR) in 2014 and is nearly finishing a M.Sc. degree from the UGR. He is currently a Ph.D. Student at the Computer Architecture Department at UPC BarcelonaTech. His research interests are in the field of Software-Defined Networks, network measurement and traffic monitoring, focusing in the identification of applications with techniques of machine learning and DPI.



Pere Barlet-Ros (http://people.ac.upc.edu/pbarlet) received the M.Sc. and Ph.D. degrees in Computer Science from UPC BarcelonaTech in 2003 and 2008 respectively. He is currently an associate professor with the Computer Architecture Department and senior researcher with the Advanced Broadband Communications Center of the UPC. He has recently co-chaired the International Workshop on Traffic Monitoring and Analysis (TMA) and served in the TPC of several conferences in the area of network measurements. His research interests are in the fields of network monitoring, network data analysis, traffic classification, anomaly detection, SDN measurements and online privacy.

Valentín Carela-Español is currently SDN Solutions Manager at Talaia Networks. He is also an external collaborator at the Broadband Communications Research Group that belongs to the Computer Architecture Department at the UPC BarcelonaTech. He received the Ph.D. degree in Computer Science in 2014 at UPC BarcelonaTech. His Ph.D. studies centered around the network traffic classification research field, focusing on the identification of applications in network traffic based on Machine Learning and DPI techniques and the aspects related to the application of those techniques in backbone networks.